Temporal Event-Based Neural Networks:
A new approach to Temporal Processing

June 9, 2023

brainchip
Essential AI

Contents

Introduction

Neural networks (NNs) form the basis of the technology for artificial intelligence. While almost all AI computation is being done in the cloud, AI computation is increasingly moving from the cloud to the edge for experience, privacy, and commercial reasons. As the complexity of NN models continues to increase, and compute demands skyrocket, it is becoming cost-prohibitive to deliver these more advanced intelligent services in real time from the cloud, or near-impossible to achieve this AI performance on small-form factor, fan less or portable edge devices, which are fundamental to the growth of the $1T+ AIoT market. It is, therefore, imperative to take a wholistic approach to solving this problem and explore radical new ways to reduce complexity, size, and compute requirements, without compromising on the high accuracy and performance expectations to support the transition to edge AI. In this paper, we present a superior method to efficiently process sequential data such as video, speech, and sensory data that is often encountered in edge applications.

A brief history of neural networks in Artificial Intelligence

Artificial neural networks (ANNs) were first developed to imitate the response of neurons, the cells in the brain. Neurons communicate with one another through electrical impulse signals called *spikes,* and it was believed that the information transmitted by neurons is encoded in the rate at which neurons emitted these spikes. We now know that the timing of these spikes, their temporal grouping, is more important. The first nonlinearities in ANNs, such as sigmoid functions, were inspired by the way neurons' firing saturate upon reaching their maximum firing rate; these became known as activation functions.

ANNs were augmented with the biological observations that individual neurons in the visual cortex respond to stimuli within a spatially small area of the visual field (their receptive field). Neurons responding to the same visual features cover the entire visual field with their overlapping receptive fields. Together with the fact that object recognition is translation invariant, these observations gave rise to convolutional neural networks (CNNs). An object is recognized regardless of its position in the visual field, or its location in an image.

Chapter 1
The rise of the CNN

CNNs are trained to recognize important spatial correlations in data, known as features. They extract increasingly abstract features as the data is processed layer by layer. Trained with gradient descent and backpropagation, the engines of deep learning, CNNs have dominated image classification and related tasks over the past decade. They efficiently extract spatial correlations from a static input image to map it into the appropriate classification with state-of-the-art accuracy.

However, many modern ML workflows increasingly utilize sequential data streams that contain spatio-temporal correlations, such as natural language processing (NLP) and object detection in video streams. The CNN models used in static image classification lack the capabilities to effectively use the temporal information that is present in these types of data streams.

How then do we give artificial neurons the flexibility to encode and process temporal information efficiently? One way is to provide them with an internal state that has some temporal dynamics, such as the state used in recurrent neural network (RNN) models.

Another way is to compute the present response based on a temporal convolution of a kernel over each of its many past inputs. As it turns out, these two methods are not exclusive. They can be made to be two sides of the same coin, and there are methods to transform from one representation to the other. This is significant because this means that the temporal convolution over an input may be transformed into a recurrent process ingesting the sequential data one timestep at a time.

Addressing the limitations of the CNN

Researchers in the field of machine learning have been experimenting with both temporal convolution and internal state approaches to efficiently incorporate temporal or sequence information. For applications that require only temporal information processing, like NLP or other sequence prediction problems, researchers have turned to RNNs such as long short-term memory (LSTM) and gated recurrent memory (GRU) models (see Figure 1). More recently, these RNNs have been supplanted by Transformers. For applications that require both spatial and temporal processing, researchers have experimented

The rise of the CNN

with performing 3D convolutions that combine 2D spatial convolution found in static image classification with a 1D temporal convolution, while other researchers have combined 2D spatial convolution with state-based RNNs such as LSTMs or GRUs to process the temporal information components with models such as ConvLSTM.

However, each of these approaches comes with significant drawbacks. Combining 2D spatial convolutions with 1D temporal convolutions is computationally expensive and is thus not appropriate for efficient low-power inference, although much effort has been expended to reduce this cost.

A core issue with RNNs is the excessive application of nonlinear operations at each timestep, which exhibits two major drawbacks. First, nonlinearities force the network to be sequential in time, meaning it cannot efficiently leverage parallel processing during training.

Second, since the applied nonlinearities are ad-hoc, without any theoretical guarantee of stability, it is not possible to train these networks or perform inference over long periods of sequential data. These limitations also apply to models that combine 2D spatial convolution with RNNs to process spatial and temporal information, such as the ConvLSTM models mentioned above.

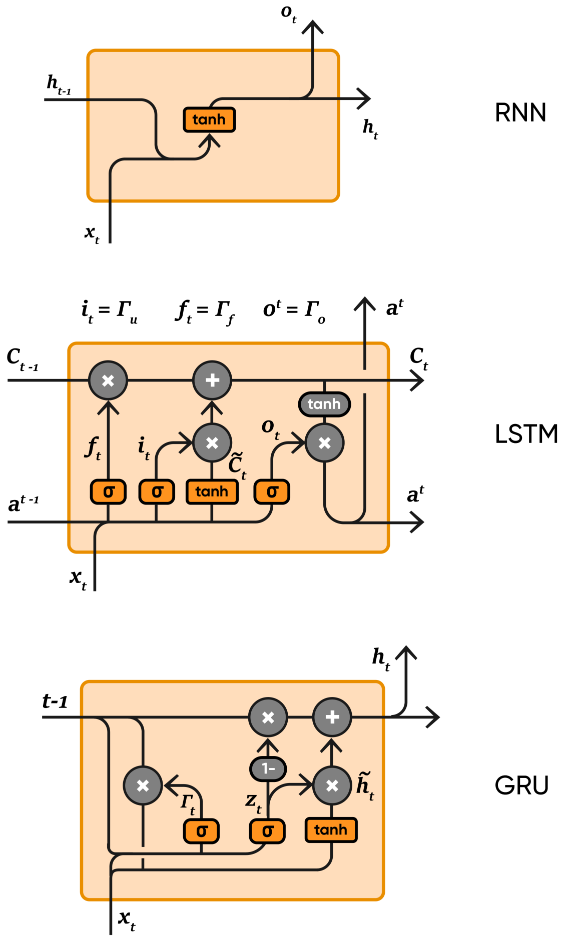


Figure 1. Diagrams detailing the important components of RNNs (top), LSTMs (center), and GRUs (bottom).

(from https://doi.org/10.1080/03610918.2020.1854302)

Chapter 2
The birth of the Transformer

Transformers were developed to find networks to replace RNNs that were a lot more efficient to train by utilizing the parallelization of training hardware in language modeling, machine translation, and question answering. In the authors' words, "the Transformer requires less computation to train and is a much better fit for modern machine learning hardware, speeding up training by up to an order of magnitude."

(https://ai.googleblog.com/2017/08/transformer-novel-neural-network.html)

Transformers were initially described as being more capable of connecting input items farther away from each other than RNNs. However, recently a new class of RNNs has been able to solve long-range arena (LRA) problems beyond the capabilities of the best Transformers (see how the LRA performance score is lower than 55 for the Transformers in Figure 2). These RNN networks (with names like S4, S4-LegS, S4D-LegS, and S5) are based on state-space equations, which have been used in engineering for modeling system dynamics, based on simplifications of physical equations closely reflecting the world's physics. Such networks exhibit an LRA performance score of 85 and higher and have succeeded in finding a solution on Pathfinder-X (Path-X) where the best Transformers have so far failed, as shown in Table 1.

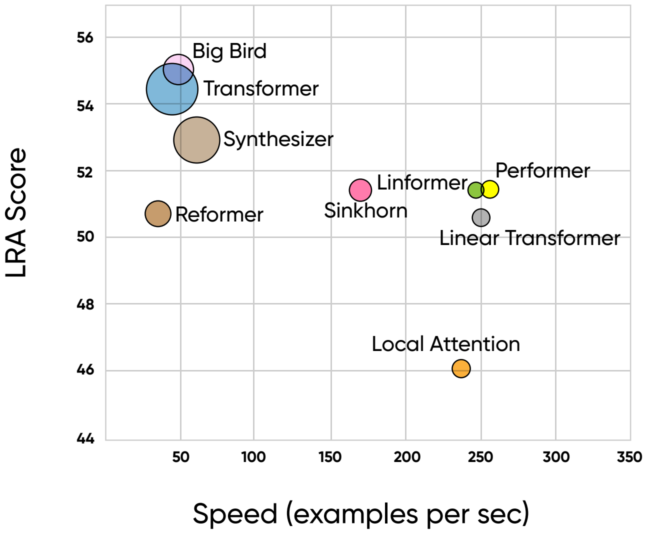


Figure 2. Long Range Arena (LRA) performance score (y axis) vs inference speed (x axis) and memory footprint (circle size) for different variations of Transformer models. The highest score is around 55. State-space models in Table 1 achieve quite a bit more at around 85 and above.

(from https://doi.org/10.48550/arXiv.2011.04006)

The birth of the Transformer

Model (Input length)	ListOps (2,048)	Text (4,096)	Retrieval (4,000)	Image (1,024)	Pathfinder (1,024)	Path-X (16,384)	Avg.
Transformer	36.37	64.27	57.46	42.44	71.40	X	53.66
Luna-256	37.25	64.57	79.29	47.38	77.72	X	59.37
H-Trans.-1D	49.53	78.69	63.99	46.05	68.78	X	61.41
CCNN	43.60	84.08	X	88.90	91.51	X	68.02
Mega($\mathcal{O}(L^2)$)	**63.14**	**90.43**	91.25	**90.44**	**96.01**	97.98	**88.21**
Mega-chunk ($\mathcal{O}(L)$)	58.76	90.19	90.97	85.80	94.41	93.81	85.66
S4D-LegS	60.47	86.18	89.46	88.19	93.06	91.95	84.89
S4-LegS	59.60	86.82	90.90	88.65	94.20	96.35	86.09
Liquid-S4	62.75	89.02	91.20	89.50	94.8	96.66	87.32
S5	62.15	89.31	**91.40**	88.00	95.33	**98.58**	87.46

Table 1: Performance on the long range arena (LRA) benchmark tasks. X indicates that the network "performs no better than random guessing." Bold and underlined scores indicate highest and second highest performance, respectively. Path-X is Pathfinder-X, Long-Range Spatial Dependencies with Extreme Length. The best Mega model has the Transformer's complexity of the order of the square of the sequence length $\mathcal{O}(L^2)$ instead of the order of the sequence length $\mathcal{O}(L)$ for S4 and S5 networks.

(Table 1 from https://doi.org/10.48550/arXiv.2208.04933)

Chapter 3
TENN: It’s about Time

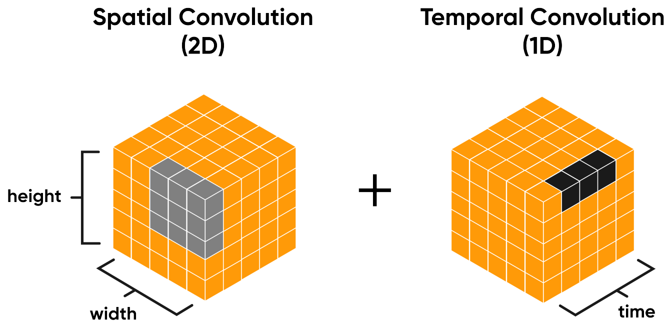
BrainChip developed its version of temporal networks organically due to its foundation in digital event-based neural networks (ENNs). An explicit temporal convolution capability has been added in the Temporal Event-based Neural Networks, or TENNs, which efficiently combine spatial and temporal convolutions.

Unlike standard CNN networks that only operate on the spatial dimensions, TENNs contain both temporal and spatial convolution layers. They may combine spatial and temporal features of the data at all levels from shallow to deep layers. In addition, TENNs efficiently learn both spatial and temporal correlations from data in contrast with state-space models that mainly treat time series data with no spatial components. Given the hierarchical and causal nature of TENNs, relationships between elements that are both distant in space and time may be constructed for efficient continuous data processing (such as video, raw speech, and medical data). *Causal* means using previous processed values of the time series to estimate current or future values.

The parameters of the temporal kernels can be optimized during training so that the shapes of the temporal kernels are driven by the data and not limited to some a priori definition.

Every temporal layer of a TENN consists of a temporal convolution between the temporal kernels and the inputs, which is a linear operation that works similarly to standard spatial convolutions in CNNs. Therefore, the temporal convolution mode of the network can be trained similarly to how a CNN is trained, efficiently utilizing modern parallel processing architectures and libraries. The training can be done using standard optimization algorithms such as ADAM.

The temporal convolution mode also guarantees the stability of the network given well-behaved temporal kernels. In addition, TENNs offer parameter representation that is more efficient in training and storage requirements than the ubiquitous weight representation used in most ANN/CNN.



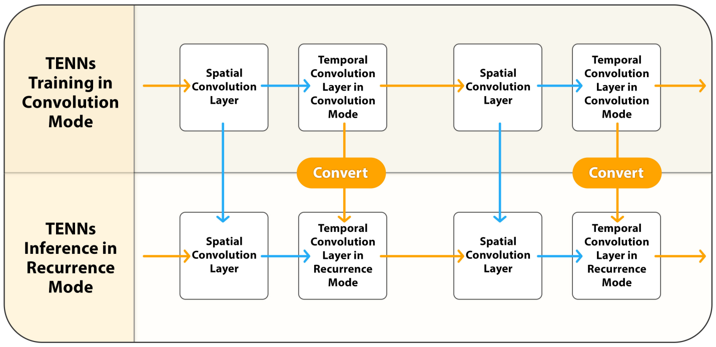
TENN: It's about Time

The parameter representation makes TENNs robust to changes in spatial and temporal scales and thus enables it to handle changes in input data resolution and temporal sampling rate.

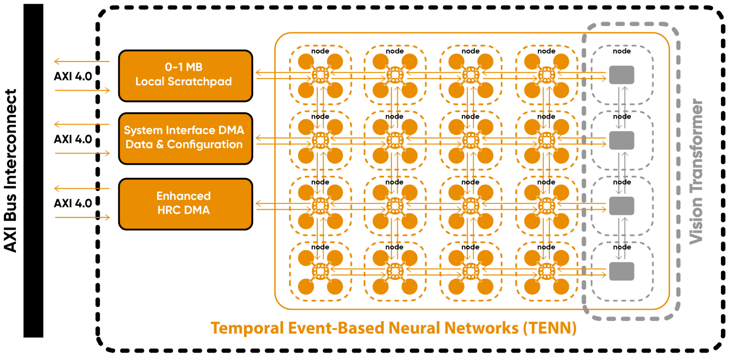
TENNs temporal operations can be configured to operate either in temporal convolution mode ("convolution mode") or as a recurrent temporal operation ("recurrent mode"). The temporal convolution layers can be naturally converted into equivalent recurrent layers for efficient online inference. This is especially useful for mobile devices and edge computing, where it is costly to keep a large memory buffer (time window) of past inputs to perform temporal convolutions at every timestep. The resulting network is more efficient with temporal operations configured as recurrent layers, performing efficient online inference over spatiotemporal data streams. Thus, TENNs benefit from efficient training on parallel hardware (such as GPU and TPU), and from the compactness of recurrence for inference at the edge.

Contrary to the large neural networks running on a cloud server that handle temporally unrelated data, the edge is often focused on receiving a continuous stream of the same data (video surveillance, self-driving video, medical vital signs, speech audio). This allows neural networks to take full advantage of the redundancies in space and time, and to develop sparse data representations that permit them to run AI hardware with efficient memory and efficient power use at the edge.

akida TENN OPERATION


TENNs Training in Convolution Mode
Spatial Convolution Layer
Temporal Convolution Layer in Convolution Mode
Spatial Convolution Layer
Temporal Convolution Layer in Convolution Mode
Convert
Convert
TENNs Inference in Recurrence Mode
Spatial Convolution Layer
Temporal Convolution Layer in Recurrence Mode
Spatial Convolution Layer
Temporal Convolution Layer in Recurrence Mode

AKIDA TENN & (Optional) Vision Transformer Mesh



Chapter 4
Results and Summary

TENNs have shown state-of-the-art (SoTA) accuracies with orders of magnitude fewer parameters and fewer multiply-accumulate (MACs) operations per inference than previous networks (see Tables 2 to 5). On the SC10 subset of the Speech Commands dataset (Table 2) (https://doi.org/10.48550/arXiv.1804.03209), TENNs achieve close to SoTA results at a fraction of the compute (MACs/sequence) and number of parameters. For comparison, the performance of a CNN specifically designed for raw speech, the discriminator from the WaveGAN model is presented among other convolution (CKConv), and transformer networks (Transformer, Performer) (from https://doi.org/10.48550/arXiv.2111.00396).

Near SoTA Results for SC10 (10-Class Raw Audio Speech Commands Classification)

Model	SC10 raw classification accuracy	Number of Parameters (million)	Billion MACs/sequence
S4 (SOTA)	98.32	0.3	44.8
TENN$_{16}$	**98.15**	**0.085**	**0.091**
TENN$_{128}$	**97.12**	**0.052**	**0.019**
WaveGAN-D	96.25	26.3	13.76 *
CKConv	71.66	0.1	230 *
Performer	30.77	~12500 *	~6.3 *
Transformer	X	~25000 *	~3100 *

X means not possible to train or failed to do better than random guessing.
* These are estimates based on available code and publication information

Table 2: Near State-of-the-Art Results for the 10-Class Raw Audio Speech Commands Classification. Accuracy percentages of TENN$_{16}$ and TENN$_{128}$ indicate that they achieve near SoTA results but with a fraction of the compute and number of parameters. The numbers 16 and 128 indicate the amount of subsampling performed in TENN, further reducing the number of parameters and compute. WaveGAN indicates comparing to a baseline CNN specifically designed for raw speech, the discriminator from the WaveGAN model. In contrast TENN is a general architecture, not specifically designed for raw speech.

(From https://doi.org/10.48550/arXiv.2111.00396).

Results and Summary

Near SoTA Results for vital signs prediction (BIDMC dataset)

Model	Respiratory Rate	Heart Rate	SpO2	Number of Parameters (million)	Billion MACs/seq
S4 (SOTA)	0.247	0.332	0.090	0.3	11.2
TENN$_4$	**0.352**	**0.392**	**0.155**	**0.084**	**0.080**
TENN$_{16}$	**0.391**	**0.472**	**0.251**	**0.044**	**0.014**
UnICORNN	1.06	1.39	0.869	0.135	0.540 *
CKConv	1.214	2.05	1.051	0.1	14.4 *
LSTM	2.28	10.7	-	0.064	~0.26
Transformer	2.61	12.2	3.02	~6300	~197

* These are estimates based on available code and publication information

Table 3: Results for vital signs prediction on the BIDMC dataset expressed as the root mean square error (lower is better). TENN$_4$ and TENN$_{16}$ achieve near SoTA results, but with a fraction of the compute and number of parameters. Compared to UniCORNN, the results indicate that TENN$_4$ has a 95% confidence interval that is 3x smaller in its prediction. The numbers 4 and 16 indicate the amount of subsampling performed in TENN, reducing the number of parameters and compute.

(From https://doi.org/10.48550/arXiv.2111.00396; https://doi.org/10.48550/arXiv.2110.13985).

On the vital signs prediction BIDMC dataset (Table 3) (https://tinyurl.com/2p8ps2rc), TENN is doing much better than the next best results (UnICORNN) and almost at SoTA (S4), but with far fewer parameters and compute (MACs/sequence). From our analysis so far, the TENN is substantially better on the event-based Prophesee 1 Megapixel Automotive Detection Dataset at a fraction of the compute and number of parameters compared to the other networks (Table 4). Finally, the same TENN architecture may not only be used on event-based data but can be directly used on frame-based video data as well. For Video Object Detection on the KITTI Dataset, TENN can match spatial CNNs even for color frames with orders of magnitude fewer compute and number of parameters compared to other networks (Table 5).

Results and Summary

Event-Based Object Detection — Prophesee 1 Megapixel Automotive Detection Dataset

Network	mAP %	Parameters (million)	Billion MACs/s
TENN	**56**	**0.167**	**44.6**
Events-ConvLSTM	43	24.1	1348 *
Gray-RetinaNet	43	32.8	2482 *
E2Vid-RetinaNet	25	43.5	> 2482 *
Events-RetinaNet	18	32.8	2338 *

Table 4: Object Detection on the event-based Prophesee 1 Megapixel Automotive Detection Dataset. TENN is the State of the Art at a fraction of the compute and number of parameters compared to the other networks. mAP is the mean Average Precision, a metric used to evaluate object detection models; it falls between 0 and 1, with 1 being the best.

https://tinyurl.com/mryddn7n

Note: The number of inferences per second for all models in Table 4 was 100 inferences per second (IPS).

* These are estimates based on available code and publication information

Frame-Based Camera Video Object Detection - KITTI 2D Object Detection

Network	mAP %	Parameters (million)	Billion MACs/s
TENN	**57.6**	**0.165**	**13.1**
SimCLR (ResNet-50)	57.5	26	82
RGBD Fusion (YOLOv2)	48.2		349

Table 5: Video Object Detection on the KITTI Dataset. TENN can match spatial CNNs even for color frames with orders of magnitude fewer compute and number of parameters compared to other networks. mAP is the mean Average Precision, a metric used to evaluate object detection models; it falls between 0 and 1, with 1 being the best. KITTI:

(from https://www.cvlibs.net/datasets/kitti/
SimCLR: https://www.lightly.ai/datasets
(Kitti 2d Object Detection Factsheet from Lightly
RGBD Fusion:
https://www.mdpi.com/1424-8220/19/4/866)

Note: The inferences per second for all the models in Table 5 was 20 IPS.

Conclusion

In conclusion, TENN clearly demonstrates a very efficient and innovative way to achieve highly accurate models to support sequential data use cases, such as with video and time series data. By making more efficient use of sequential data, and eliminating the need for separate preprocessing, TENNs provides simpler network and chip architectures for smaller, cheaper, lower power devices without compromising on the high accuracy and performance expectations to support the transition to edge AI. Because TENNs can be trained as convolutional models utilizing the same parallel training pipelines in use today, which substantially reduce the training time compared to the sequential training of traditional recurrent models, TENN facilitates a wider adoption and a faster speed to market than past traditional recurrent networks ever could. Designers can therefore, more seamlessly incorporate TENNs into their current training methodologies and workflows.

BrainChip's 2nd Generation Akida IP provides full support for TENN’s exciting innovation, to enable truly intelligent end point devices for advanced applications like video object detection, audio and healthcare applications with orders of magnitude better performance than was previously possible. We believe this will empower designers to develop radically efficient, intelligent edge solutions, which can, in addition, benefit from the set of Akida's other unique abilities, such as on-chip learning at the edge.

brainchip

Essential AI